


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published before October 2003

Terms used

Found 51 of 147,137

[instrumenting](#) [profile](#) [code](#) [bytecode](#) [execution](#) [paths](#)

Sort results by

☒ relevance

[Save results to a Binder](#)
[Try an Advanced Search](#)

Display results

☐ expanded form

[Search Tips](#)

 Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 51

Result page: [1](#) [2](#) [3](#) [next](#)Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [A framework for reducing the cost of instrumented code](#)



Matthew Arnold, Barbara G. Ryder

 May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.78 MB\)](#)
 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Instrumenting code to collect profiling information can cause substantial execution overhead. This overhead makes instrumentation difficult to perform at runtime, often preventing many known *offline* feedback-directed optimizations from being used in online systems. This paper presents a general framework for performing *instrumentation sampling* to reduce the overhead of previously expensive instrumentation. The framework is simple and effective, using code-duplication and *coun* ...

### 2 [Mobile code: Anomaly intrusion detection in dynamic execution environments](#)



Hajime Inoue, Stephanie Forrest

 September 2002 **Proceedings of the 2002 workshop on New security paradigms NSPW '02**

Publisher: ACM Press

Full text available: [pdf\(867.25 KB\)](#)
 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe an anomaly intrusion-detection system for platforms that incorporate dynamic compilation and profiling. We call this approach "dynamic sandboxing." By gathering information about applications' behavior usually unavailable to other anomaly intrusion-detection systems, dynamic sandboxing is able to detect anomalies at the application layer. We show our implementation in a Java Virtual Machine is both effective and efficient at stopping a backdoor and a virus, and has a low false posi

**Keywords:** Java, anomaly detection, dynamic sandboxing

### 3 [Partial method compilation using dynamic profile information](#)



John Whaley

 October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and**